

Tangible and Beneficial Course Objective and Teaching Materials Proposed for Management- and Information Science-Oriented Students(Part I)

Hiroshi NOTO
Khaled BUGRARA

Contents

1. Introduction
2. Basic Concepts
3. Course Objectives
4. Management of Class and Laboratory
5. Contents and Teaching Materials

1. Introduction

In this article we present innovative approaches in teaching software application development to the students who major in management and information sciences in undergraduate schools, and the students who major in information systems in graduate schools.¹⁾

Conventional lectures and practices for those students are apt to have either “programming technique-biased”, “information technique-biased” or “management-biased” tendencies, depending on the lecturers’ majors themselves. In other words the teaching methods in software development courses have been relied on neither on the proper characteristics of software development per se nor the socio-technical needs of software from clients and society.

On the other hand, the software development courses usually include rather a wide range of students in their interests and concerns, from management-oriented students to engineering-oriented students, moreover with their preparation levels varying. Many of the students in the undergraduate schools have never even written programs in a programming language before. In addition to that engineering-oriented students tend to have little interest in programming and information science-oriented students tend to have little interest in business and management issues.

The students who graduate from colleges or universities having finished software development courses are, sooner or later, unavoidably compelled to compete with other persons of the company or rival companies in various categories of businesses and get ahead in the business fields. More specifically, for example, they have to acquire knowledge and skills to construct their own business models, and realize their models as management

Key words: Business Model Construction, Management Information System Development, NetBeans IDE (Integrated Development Environment), UML (Unified Modeling Language), Logical Thinking Visualization

information systems on the computer or on the Internet.

The students should be able to deal with both business model construction and management information system development equally well. What do the efficient and effective teaching methods look like to get the students to understand how to make a business model that meets a company's strategy and requirement systematically, efficiently, fast and within a limited period of time? What kinds of teaching materials could we prepare to encourage the students to develop a management information system by implementing the business model efficiently, fast and within a limited budget?

To this end, we make full use of methodology both in business model construction and in management information system development²⁾; we have to keep track of the newest trend of the software development tools and get the best language processor and the integrated development environment on the computer and on the Internet ready as our system developing platform. Here we have selected NetBeans IDE³⁾ (Integrated Development Environment) in Java⁴⁾ for software development and UML⁵⁾ (Unified Modeling Language) for business modeling⁶⁾. The characteristic feature of NetBeans and UML are their visualizing capability of the model structure and the system architecture, and the operability based on their object-oriented programming concept and its methodology.

On the platform mentioned in the previous paragraph we could carry out practices and work without going deeply into programming knowledge, grammar or techniques, could make full use of our tools on the platform without being bothered with huge data or multi-functional procedure pertaining to and traversing to organizational sections in business models, and could design screens and their sequences that deal with those data and procedures modularly and visually as the objects in the language processing system. The visual practices and work of building and manipulating business models on such a platform makes the students interested in and feel challenged with software development itself.

In our innovative approaches, then what are the actual contents of the teaching materials all about on the Java platform: how to get them ready; how to deliver lectures on software development and how to run the students' practices of constructing business models and creating management information systems; and how to manage our courses as a whole and how to educate and empower our students.

In §2 our basic concepts of software development are set forth, referring to what is missing in conventional software or information system development education in the software development courses in graduate schools and undergraduate schools. The objectives of our development courses are presented in §3. An example of how to run our class and laboratory is elaborated in §4. The contents prepared and the teaching materials used in our courses are explained in §5. In §6 the questionnaire about our courses have been ready for the students' feedback. The conclusion of this article is summarized in §7.

2. Basic Concepts

Our management- and information-oriented courses should facilitate career transformation for students with different information technical backgrounds. In our graduate school and undergraduate school, the programs of our courses mix academic and corporate experience as a way of enabling our students to develop both their professional knowledge and skills.

Our program targets the skill sets important for the corporate world where the focus is on specification requirements, design, implementation and ongoing operations of business applications. Our program should give students the methods and tools that make it practical for them to build better software systems that meet business needs, expectations and desires. Students master the skill sets needed to make well-informed decisions about what techniques work best in given situations in building business systems. With those knowledge and skills reorganized in each student they are to become to acquire the ability to communicate and interact with business owners, end-users and other stakeholders.

We empower students to cater to the information engineering and management needs of the corporate world. We offer courses that deepen academic knowledge while building the skill sets useful for their expected workplaces: a reasonable combination that integrates academic learning with practices and workplace experience (mostly graduate schools). The courses of software development should strike the right balance between management engineering and information science.

Our program should be unique in its approach to software design and development. Software engineers are strong technically but often weak when it comes to business and management skills (soft skills). Conversely business strategy and business project carried out based on business persons' decisions have mostly failed. This is because business persons have not enough knowledge that meets practical decisions about technology and how it fits into the present business environment.

Software development is carried out with the knowledge of programming and business management. The former requires a lot of rigorous and logical thinking, whereas the latter deals with the complexity of the underlying business concerned. In order to facilitate those knowledge and skills, we need to vividly show the students the depth and rigor of abstract thinking and complex logic, and the importance of a lot of practices. The core of our innovative teaching is "design abstraction and complexity."²⁾ Design is to visualize managing abstraction and complexity and this work requires a lot of practices. It is indeed in our software development courses that we convey the double idea of abstraction and practicality. Since the power behind design and architecture is conceptual thinking, the students' getting progressed in design and architecture is a necessary condition for their effective team collaboration and their appeal of their architectural intention and concepts. The students thereby are to solve complex and practical logical problems that require a lot of abstract thinking and programming.

In practical situations, however, such as in our class rooms and in our laboratories, it

is extremely difficult for the students to deal with software development problems mentioned above. Still it should be possible with our proper and patient guidance that the students move to higher levels of abstraction by practicing and experiencing design problems as comprised of subdivided pieces, if necessary, yet visual and self-contained and concrete systems that they build by themselves.

We will make our program not be an MBA or a traditional Business Administration course (BAC) or not be computer science but just somewhere between where a solid software engineering foundation is interlocked with strong business and IT management knowledge². This surely creates unique skill sets for our students: a strong technical and engineering foundation coupled with strong management and business skills. It is these skills that are a must for building systems which people can use effectively and that certainly are outside the reach of MBA or BAC and computer science.

The essence or core part of our courses lies in business modeling and system modeling performed visually. Formal models enable software engineers to build systems that facilitate communication in a way that is independent of specific vendor technologies and application areas. When engineers make a model of the business system, the components should be selected very carefully, ensuring that the most basic foundational components and distinct features are specified and woven into the system. It is these formally defined relationships that provide the meaning to the components chosen and that provide the operations of the overall system.

In our courses we make full use of our tools that deal with the basic components of our business model and our system architecture where we manipulate designing screens and their sequences (i.e. navigation of screens) modularly and visually as objects. Our visual work is carried out with NetBeans IDE (Integrated Development Environment) on Java platform. One of the good frameworks to facilitate the visual modeling is UML (Unified Modeling Language). We introduce this framework into our courses according to their necessities in our undergraduate school.

3. Course Objectives

We present the case studies of the courses of two schools that the present authors are in charge of: Graduate School of Engineering and Department of Management and Information.

◇ Graduate Schools

Department of Information Systems, Graduate School of Engineering,
Northeastern University.

Here we take one course as a typical example of visual software development that is given in Department of Information Systems, since this course can give a hint as to and provide inspiration for how to run several courses in our undergraduate school.

Course: Application Engineering and Development (2008-2010)²⁾

<Objective>

The students develop a complete application, no matter how small the scale is, useful for an end user. The activities include design and programming.

<Lesson Overviews>

- ✚ Outline the key deliverables for the day. These deliverables are described in visual form so that the students can understand the problem. The visual user interface fully articulate what the end users want.
- ✚ The lecture sessions rely on visual user interfaces and ways to abstract the visual screens into information models or conversely to realize the information models on the visual screens.
- ✚ In the lab work the students are encouraged to follow the instructions to precisely implement a complete application presented in the class. At this point the students feel empowered since they are now able to accomplish some self-contained work on their own.
- ✚ The homework assignment is usually a close replica of what they develop in the class, with a few twists to make them feel like they are challenging nontrivial problems.

◇ **Undergraduate Schools**

Department of Management and Information, School of Economics,
Hokusei Gakuen University.

Courses (2008)

- Software Development I (Junior: 1st semester)

<Objective>

“Structured Method with CASE tool” in Software Development in Practical Business Handling (Management).

<Lesson Overviews>

- Structured Design, Structured Programming
 - ✚ Specific Management System
- Data Model, Process Model
 - ✚ Definition of Data Structure
 - ✚ Event Diagram, Action Diagram
- CASE (Computer Aided Software Engineering)
 - ✚ Structured Design, Structured Programming
 - ✚ Code Generation

<Expected Effects>

- Decrease in the period of Programming Process
- Decrease in the number of Workers (Personnel)

■ Software Development II (Junior: 2 nd semester)

<Objective>

“Structured Method with CASE tool” in Software Development in Practical Business Handling (Management) (II)

<Lesson Overviews>

- CASE (Computer Aided Software Engineering)
- Object-Oriented Programming
- Event-Driven Mechanism
- Structured Design
- Module Design Specifications (Document)
 - ✚ Display Design
 - ✚ Graphical User Interfaces
 - ✚ Event Handling
 - ✚ Display Layout Diagram
 - ✚ Report Layout Diagram
- Code Generation
- Source Code in Windows C
- Source Code of Specific Management System
 - ✚ Arbeit Management System
 - ✚ Grades Management System
 - ✚ Inventory Management System
- File Management
- Project Management

<Expected Effects>

- Decrease in the period of Programming Process
- Decrease in the number of Workers (Personnel)
- Increase in Productivity

Courses (2009-2013)

- Software Development I (Junior: 1 st semester)
- Software Development II (Junior: 2 nd semester)

<Objective>

Management Information System development on Java IDE with Unified Modeling Language (UML) plugged in —Business Model and Object-Oriented Programming—

<Lesson Overviews>

- Object-Oriented Programming
- Java Programming Language
 - ✚ Visual interfaces in Java/Swing
 - ✚ Create and display relevant screens
 - ✚ Visual user process sequences
- Unified Modeling Language (UML)
 - ✚ Class Diagram and Entity Relationship Diagram
 - ✚ Object-Relational Mapping (ORM) between Object Model and Data Model^{5,7)}
 - ✚ Code Generation and Reverse Engineering^{5,7)}
- Implement Business Logic
- Design Layout and Printout⁸⁾

■ Application II (Senior: 2nd semester)

<Objective>

Web application on Java Platform¹¹⁾ with Unified Modeling Language (UML) plugged in —Business application with Enterprise Java Beans (EJB)⁹⁾ and MVC (Model-View-Controller) Framework—

<Lesson Overviews>

- Web Service and EJB (Enterprise Java Beans)^{14,15)}
- Three-Tier Architecture of Model-View-Controller (MVC) Framework
- MySQL and Database Tier
- GlassFish Server and Data Source¹⁰⁾
- Business Logic and Session Enterprise Java Beans^{12,13)}
- ORM Mapping between Data Model and Object Model with UML

4. Management of Class and Laboratory

For the students to learn earnestly with their concentration in the class room and to get down to the problems vibrantly with enthusiasm in the laboratory, special attention should be paid to running and managing the classes and laboratories of “Management and Information Development” courses. Itemized below are important remarks according to each category.²⁾

- We should make it clear who you are and what kind of abilities you make an effort to increase in this course

You are in between IT specialist of programming and (traditional) clerical and managerial work. Software development knowledge and programming skills are expected to acquire for the hardware-oriented students or IT devices-oriented students. Require knowledge of management information system and business model for hardware- and programming-oriented students. Require system analysis-like and business model-like way of thinking and programming skills for the students who are aiming at becoming clerical and managerial staffs (product manager and sales manager).

- Connection between lectures and practices

As for the knowledge obtained in the lecture, you can confirm and understand its meaning and role by solving the problem and find out insufficient grasp of the points of programming technique. It seems rather effective to run the class on a questions and answers dialog basis to keep the atmosphere of the class vibrant, particularly when following and memorizing a new concept and a new structure of architecture. Immediately after presenting a unit of concept of information systems and a unit of programming technique, e.g. methods, we should set forth a problem relevant to understanding the new stuffs.

- Visual explanation and visual teaching material

We put emphasis on visualization of lectures, exercises and problems so that the students are able to understand the conceptual matters and the structural characteristics of system construction and its programming. Conversely a visual way of thinking and a visual explanation sometimes elucidate the lack of logic or the inadequacy or the inconsistency of the original conceptual explanation or the logical errors of the original system architecture.

◇ e.g. Sequence of Screens or Navigation of Screens

- Make clear the amount of materials ready at the very beginning

It is important to make ready the whole materials before the students at the very beginning of the lectures and the practices so that the students can evaluate and grasp the grade in difficulty of the lectures, the amount of their work and the time schedule of their practices.

- Elucidate the core of the problem

Curtail verbose or redundant or even unnecessary processes of solving the problems, i.e. always deal with the core of the problem and the essence of the system clearly. Then the students can concentrate on the most important part of

the problem or the essential structure of the system right away. One of the good ways would be to make the skeleton of the system program ready:

- ✧ e.g. Screen sequence is given the students at hand.
- ✧ Conversely ask the students only to make the screen sequence design ignoring other details.

- Solve a problem on your own

We tell a student to solve a problem on his or her own. No matter how often the students discuss about the clues to tackle the problem, it is each of the individuals who finally solve the problem, and complete and present their reports. This practice will teach a student what solving the problem is all about and what it looks like to present a report.

- Remarks on giving the problems or the assignments:

One problem should be self-contained in itself. One problem can be executed on its own as it is on the Java platform. The students thereby can feel a sort of attainment and get-it-done feeling!!

- Make the students challenge their reports or assignments according to their levels

If the problem is very difficult, get it subdivided into small pieces so that the students can keep their challenging attitudes going. We should, of course, solve the problems before giving them to the students. However, we should not present them as they were when we solved them.

- ✧ Always find out a way to make the students get down to their assignments and keep their efforts and challenges!!

- Teacher and TA's (Teaching Assistants)

TA is a mediator between a teacher and the students. He or she can answer student's question. TA can prepare teaching materials and upload them on the web instead of a teacher.

- Communicate with the students as many times as possible

Be patient to the students. We should better not evaluate a student when we are responding to a student's questions. Keep an eye on the students' feedback wherein a good hint for giving a lecture or answering to a student's questions may exist.

5. Contents and Teaching Materials

In this section we present several examples of contents and teaching materials in our

graduate school and our undergraduate school.

◇ Graduate Schools

Department of Information Systems, Graduate School of Engineering, Northeastern University. For the sake of discussion, the following course in particular relevant to our undergraduate courses is elaborated. This course covers how to leverage object-oriented techniques as well as user interface design principles to engineering multi-role business applications. Students learn how to incorporate such features as configuration management, user administration, and role-based-access control.

Course: Application Engineering and Development

Throughout in this course dealt with are the items on the portal site for sales persons and sales managers in business as the examples of teaching materials and the problems for the students. The complete contents, themes, concepts, methods, techniques and problems are summarized in Table 5-2 later.

We remark here only the important and innovative points in our contents. By solving problems one by one, the students understand and handle the concepts, methods and techniques on the Java NetBeans platform:

- Package, Class and Object, Instantiation, Event, Table, ArrayList, Hierarchical structure (Package, Class), Generic class
- Java/Swing, JFrame, JPanel, initComponents(), jTable, jScrollPane, ArrayList <ClassName>, Split Pane, CardLayout.

In **Table 5-1** business related items and their realization in Java programming or Object model are shown.

Table 5-1. The correspondence between business-related items and their realization in Java programming or Object model

Business-related items	Java programming or Object model
Catalog, Account	Directory
Multi Role	Categories
Business Model	Hierarchical structure
Screen	Interface, Modularity
Navigation of Screens	Object model, Relationship
Work Area	Event, Sequence of Screens
Role of processes	Use Case
Real time update	Dynamical plug-in

Shown below in Fig.5-1 through Fig. 5-10 are the examples of Lab problems (a prototype of User Management System) which are set forth in this course. A custom manager registers a new customer to his or her account, thereby managing the User

Account. After that the customer can login in the login screen and enter the Work Area. The aspect of the web application development is mainly dealt with in the advanced course of “Advanced Application Engineering and Development.” The course features how to design and implement large-scale software systems that support the needs of hundreds of thousands of users in the presence of failure conditions and how to engineer software systems that satisfy user requirements for fast-response times.

1) Main Menu of the User Management System



Fig. 5-1. The main menu screen of the prototype of a User Management System. Click on the [Manage Person] button in the left pane.

2) Manage Person Screen

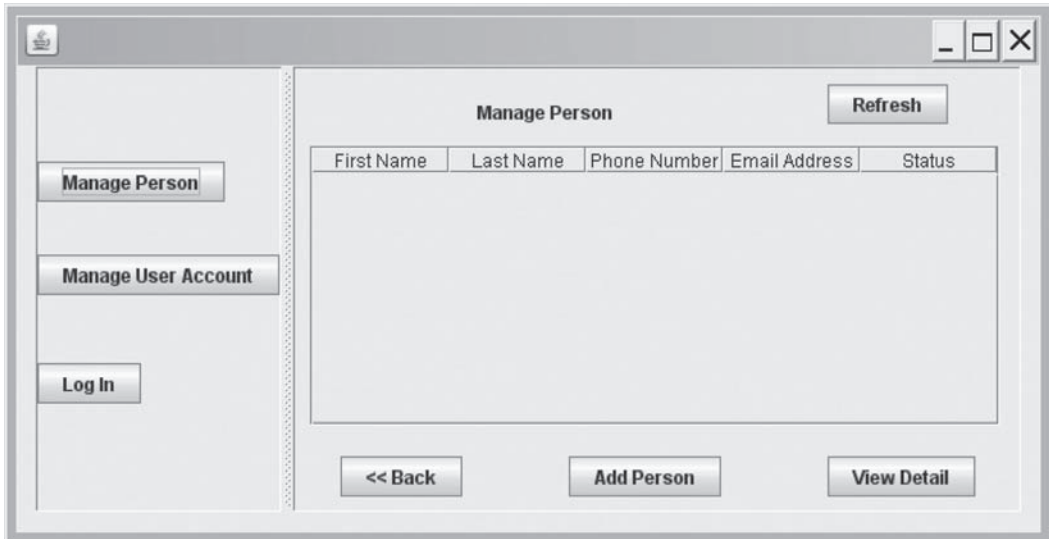


Fig. 5-2. The “Manage Person” screen appears in the right pane. Click on the [Add Person] button.



Fig. 5-3. The “Add New Person” screen appears in the right pane. Enter the fields of a new person and click on the [Add Person] button.

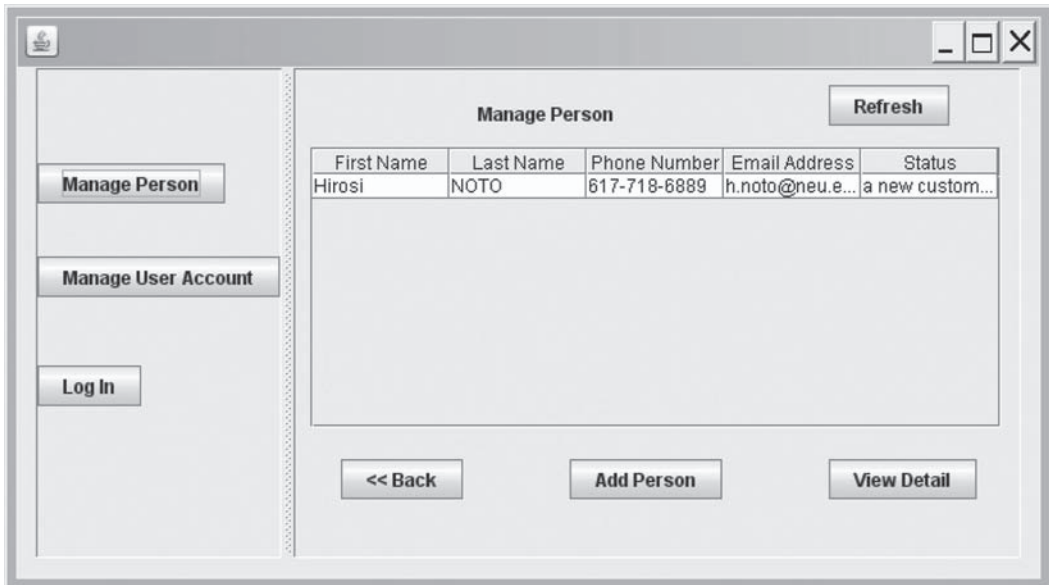


Fig. 5-4. A click on the [Manage Person] button in the left pane displays the “Manage Person” screen in the right pane with the relevant fields’ values of the new person.

3) Manage User Account Screen

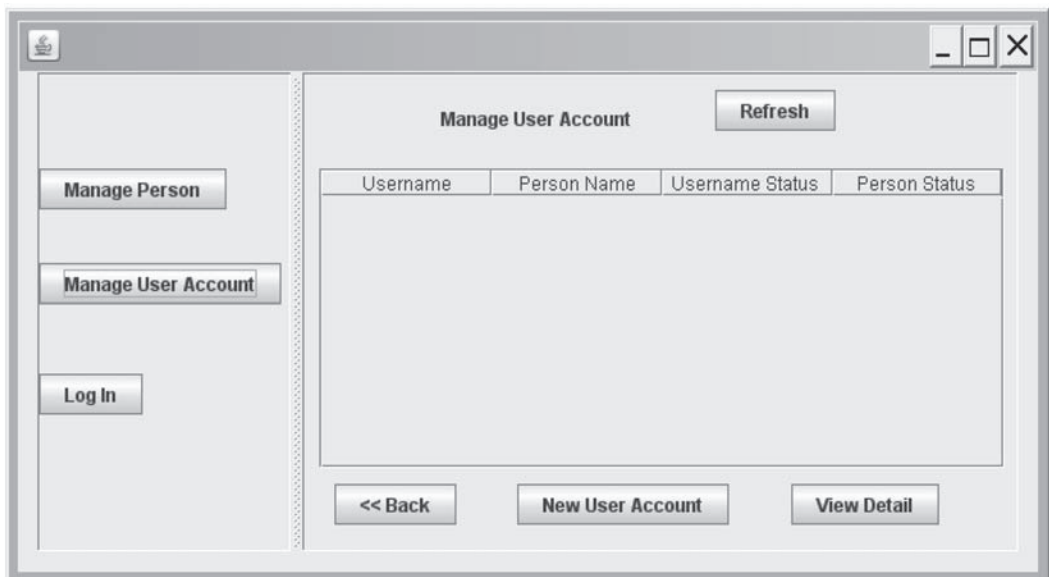


Fig. 5-5. A click on the [Manage User Account] button in the left pane displays the “Manage User Account” page. Click on the [New User Account] button.

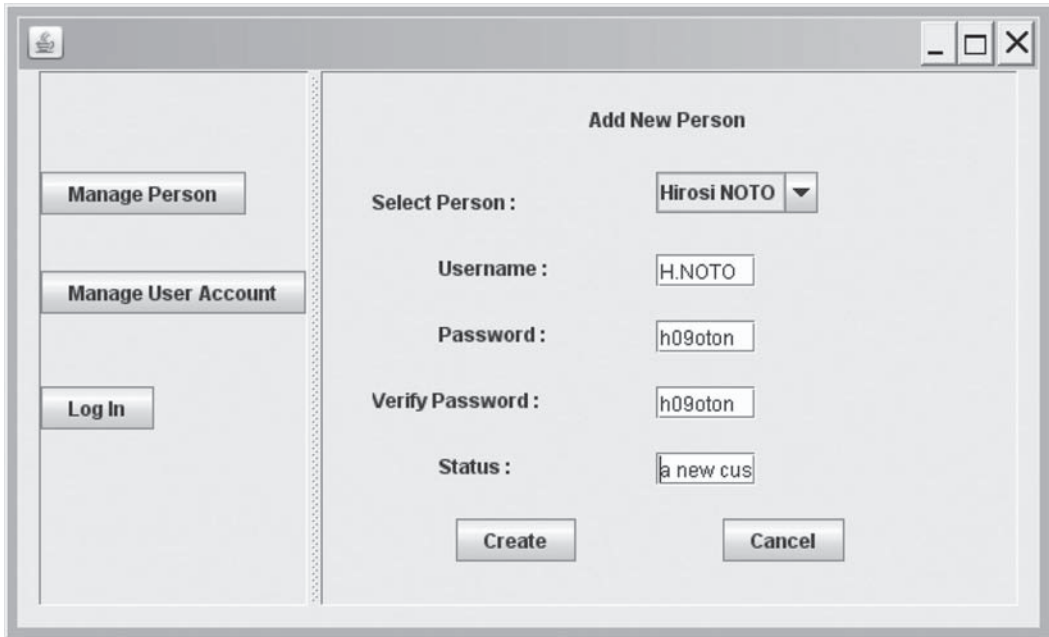


Fig. 5-6. The “Add New Person” screen appears in the right pane. Select a person in the “Select Person” list and you will be prompted to enter a Username and Password combination. Click on the [Create] button to make a new user account.

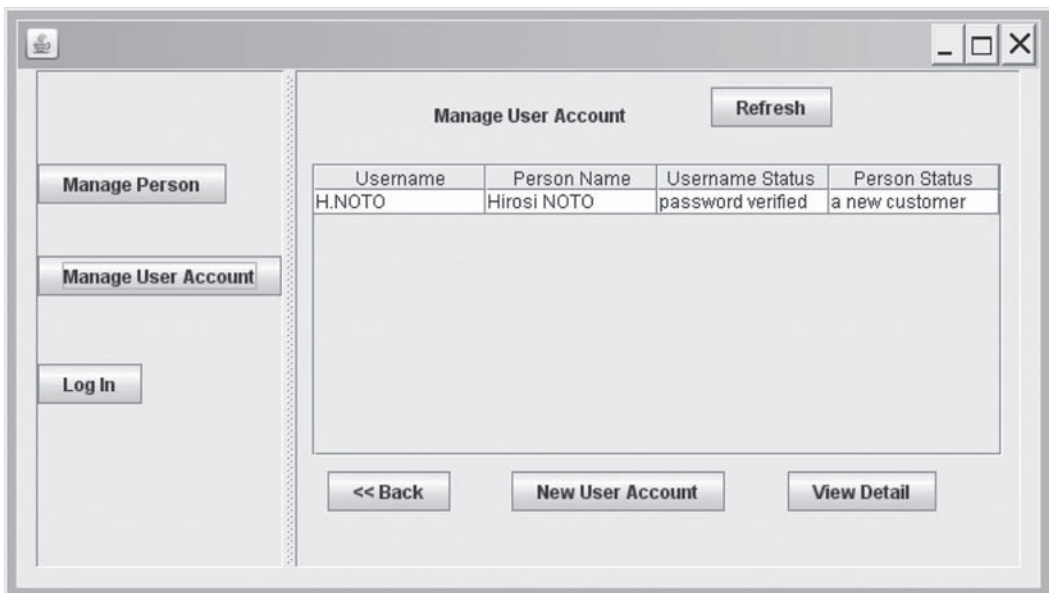


Fig. 5-7. A click on the [Manage User Account button] in the left pane shows that the created new user account is confirmed on the “Manage User Account” screen in the right pane.

4) Login Screen

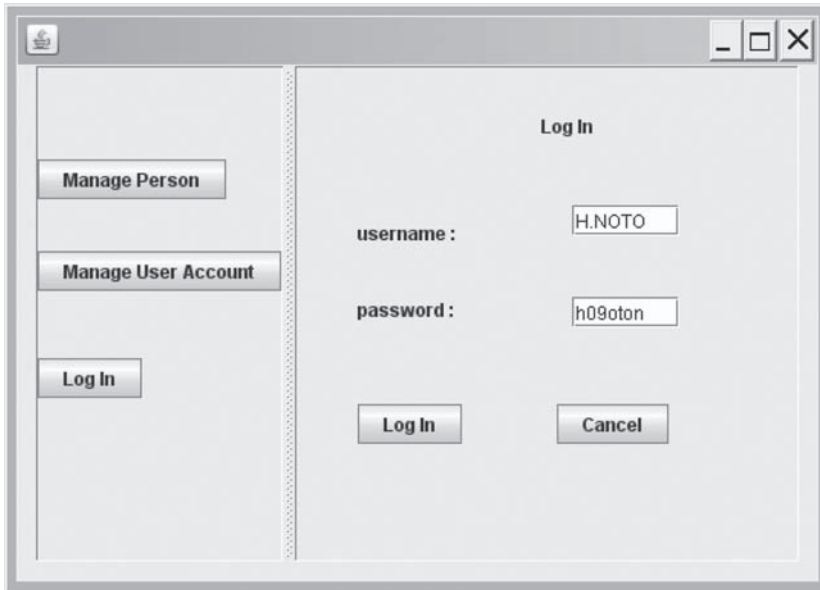


Fig. 5-8. To log in to the User Management System, click on the [Log in] button in the left pane.

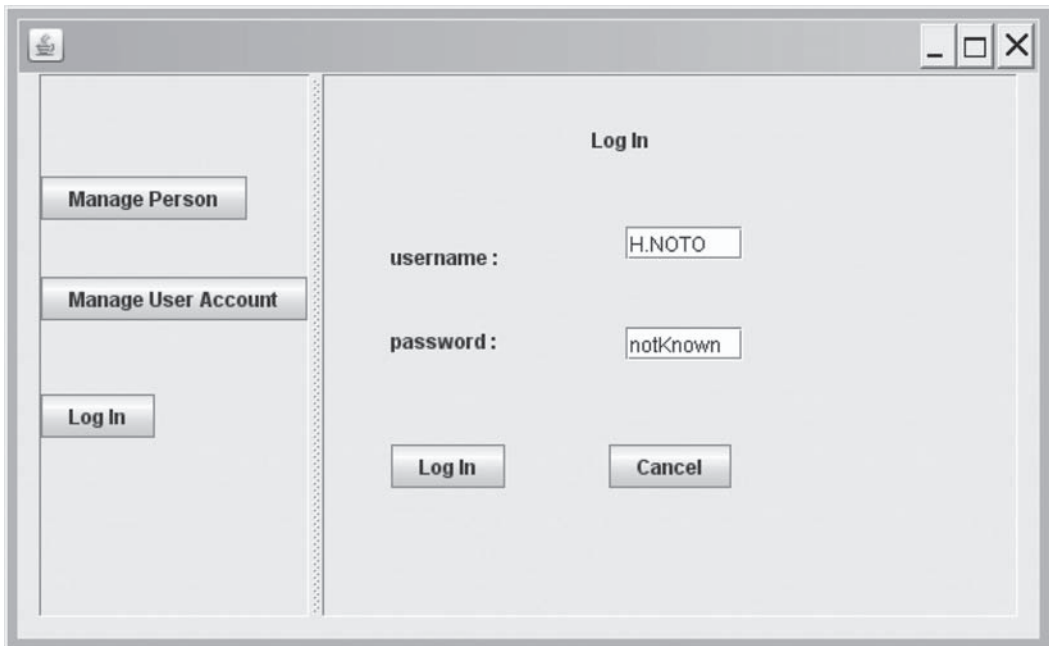


Fig. 5-9. On the “Log in” screen in the right pane, type in a username and a password in the corresponding text boxes and click on the [Log in] button.

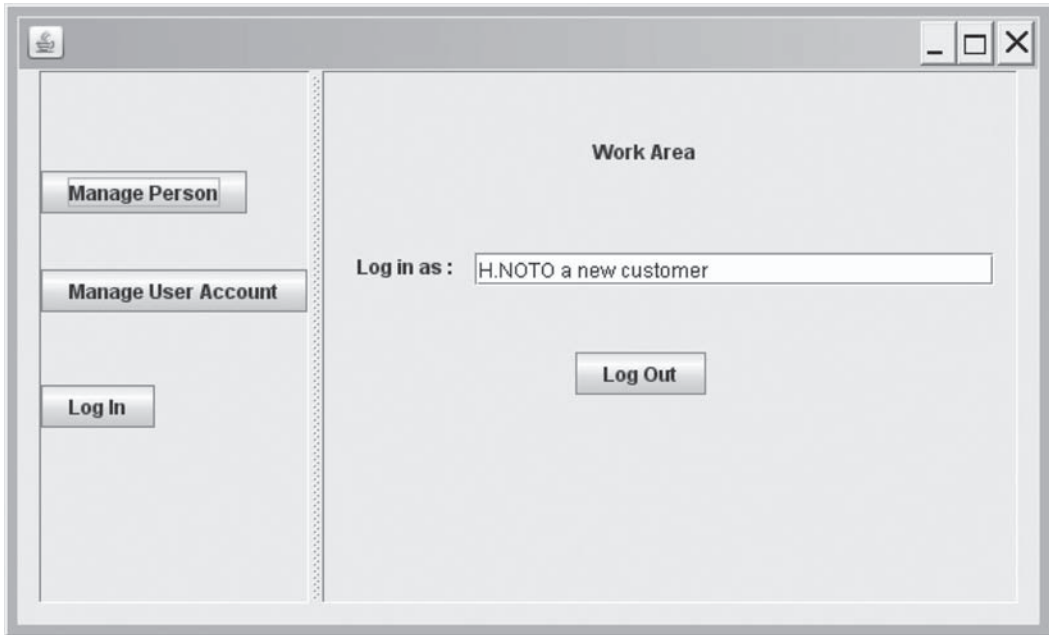


Fig. 5-10. A customer's login is accepted and the Work Area appears in the right pane.

5) Logout Screen

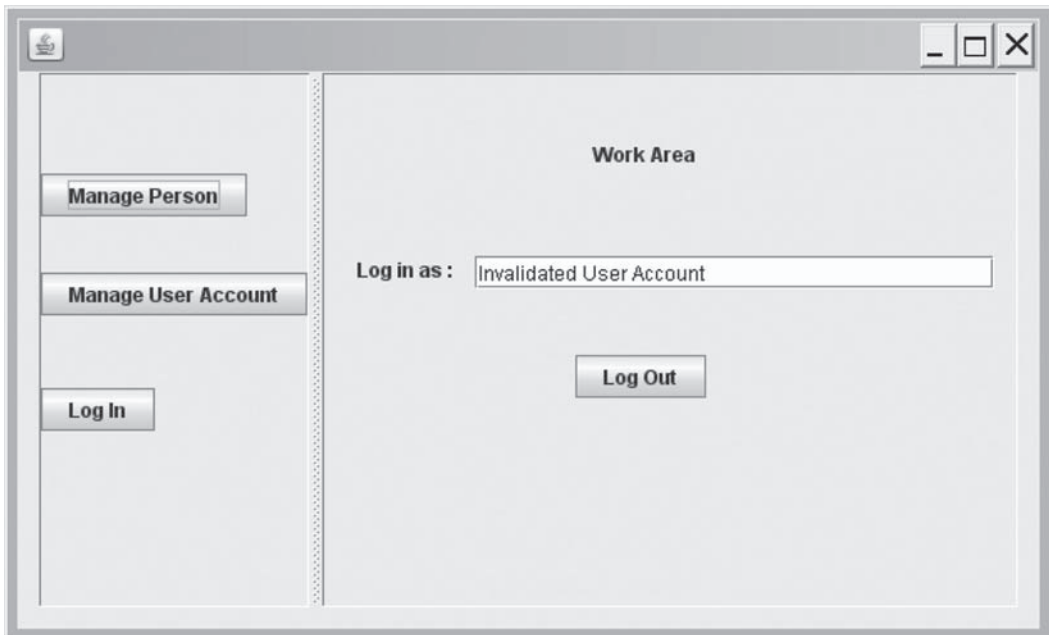


Fig. 5-11. If a username and password combination is wrong in Fig. 5-9, a customer's login is denied. Click on the [Log Out] button.

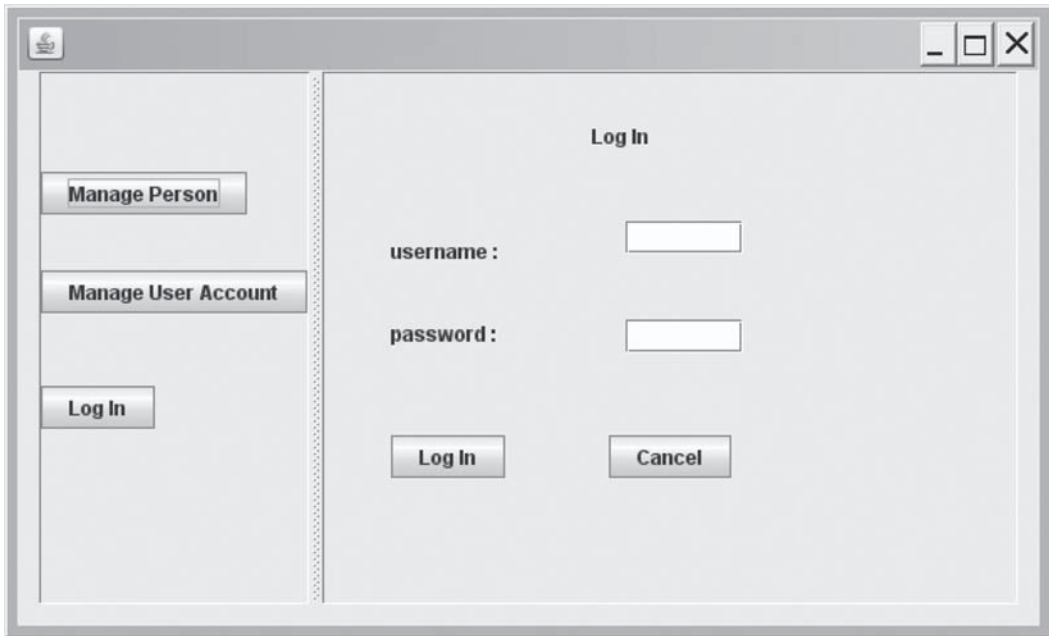


Fig. 5-12. A customer's login screen appears again.

References

- 1) One of the authors (H. N.) took sabbatical leave and stayed in Graduate School of Engineering Northeastern University for half a year from 2008 through 2009, where he was allowed to join Professor Khaled BUGRARA laboratory and was able to have valuable experiences participating in a software development class in the graduate school. He could learn a lot about business modeling and visual abstraction.
- 2) Kal Bugrara. Syllabus in Northeastern University - Graduate School of Engineering Northeastern University "Information Systems Who We Are", 2008-2009.
- 3) <http://netbeans.org/>
- 4) Mary Campione and KathyWalrath. The Java Tutorial Second Edition: Object-Oriented Programming for the Internet, Sun Microsystems, Addison Wesley, 1998.
Cay S. Horstmann, Gary Cornell. Core Java Volume I and II, eighth edition, Prentice Hall, 2008.
- 5) <http://www.visual-paradigm.com/>
- 6) Tom Pender. UML Bible, Wiley Publishing Inc., 2003.
- 7) <http://dev.mysql.com/>
- 8) <http://community.jaspersoft.com/project/ireport-designer>
- 9) <http://docs.oracle.com/javaee/>
- 10) <http://glassfish.java.net/>
- 11) Heiko Boeck. The Definitive Guide to NetBeans Platform 7, Apress, 2012.
- 12) <http://netbeans.org/kb/docs/web/mysql-webapp.html>
- 13) <http://www.oracle.com/technetwork/java/index-jsp-135995.html>
- 14) The Java EE 6 Tutorial: <http://docs.oracle.com/javaee/6/tutorial/doc/>.
- 15) Yuli Vasiliev. Beginning Database-Driven Application Development in Java EE using GlassFish, Apress, 2008.

[Abstract]

Tangible and Beneficial Course Objective and Teaching Materials Proposed for Management- and Information Science-Oriented Students(Part I)

Hiroshi NOTO
Khaled BUGRARA

Innovative approaches are presented in teaching software application development to the students who major in management and information sciences in undergraduate schools, and the students who major in information systems in graduate schools. The students should be able to deal with both business model construction and management information system development equally well. Selected in our courses is NetBeans IDE (Integrated Development Environment) in Java platform for software development with UML (Unified Modeling Language) plugged in for business modeling, in order to make full use of methodology in business modeling and in information system developing. The core of our innovative approach relies on “visualizing logical thinking”that designs abstraction and complexity. The visual practices and work of building and manipulating business models on such a platform makes the students interested in and feel challenged with software development itself. The curriculum contents and their teaching materials are set forth based on our basic concepts and methodology along with examples of laboratory work in one course of the graduate school and in several courses in the undergraduate school. (176 words)

Key words: Business Model Construction, Management Information System Development, NetBeans IDE (Integrated Development Environment), UML (Unified Modeling Language), Logical Thinking Visualization